

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

EV205822825

**Remote Computer Graphics Performance Testing**

Inventors:

Demitri Anastassopoulos  
Jason Charles Browne  
Donald Alan Page

ATTORNEY'S DOCKET NO. MS1-1561US

1      **TECHNICAL FIELD**

2      This disclosure relates to testing computers from a server computer, and  
3      specifically to graphics testing of remote client computers through a  
4      communication channel and use of a test tool application resident in the server  
5      computer.

6

7      **BACKGROUND**

8      Computers, in particular personal computers (PC), are configured with  
9      various hardware and software components. PC manufacturers are numerous, and  
10     the different PC configurations offered are even greater.

11     In some cases, a particular version of a hardware component may be used  
12    for different PC components. The behavior or performance of the particular  
13    hardware component may vary between various PC configurations, since  
14    particular PC configurations affect the hardware component differently. The  
15    hardware component may rely on a software driver (driver) that instructs the  
16    hardware component how to operate. Optimization of the hardware component  
17    within a particular PC configuration may be performed by providing a driver  
18    which factors how the hardware component operates within the particular PC  
19    configuration. Alternatively, hardware changes may be made to the hardware  
20    component.

21     An example of a hardware component is a graphics card that interacts with  
22    a PC display, PC operating system, and graphics software, where the graphics  
23    software may interact with a video adapter and a hard drive. When testing is  
24    performed on a graphics card, not only is the graphics card tested, but the various  
25    hardware and software components it interacts with are also tested.

To properly create drivers or provide hardware changes that optimize performance of the hardware component, testing is performed on the hardware component in the particular PC configuration. Typically, various editions of a driver and/or the hardware component are installed and tested for each particular PC configuration, and tests are run to determine optimal changes between driver editions and hardware component versions. In certain situations it is desirable to compare different hardware components with one another for a particular PC configuration or configurations. For example, a graphics card vendor may want to compare its graphics card with that of a competitor. In other cases, a PC manufacturer may desire to compare various graphics cards.

“Thin client” PCs lend themselves to testing using varying hardware configurations, since thin client PCs have limited software components. In particular, thin client PCs rely on a server computer to provide software applications. Therefore, the software configuration of a thin client PC remains consistent, while its hardware configuration may vary.

Various metrics may be used in testing. A particular metric is time. Tests are timed and compared to one another to determine improvements in how long a test is ran for different hardware components and/or drivers for a particular PC configuration.

A test application program that tests the hardware component may be installed on a particular PC configuration and run using a particular edition of the driver and/or version of the hardware component. This is referred to as “standalone” testing and a copy of the test application program is installed for each PC configuration. For multiple PC configurations, multiple copies of the test application program may be needed. In certain cases, a dedicated computer that

1 includes the test application may perform testing for multiple computers. There  
2 may be a need to provide test specific interconnections to physically connect the  
3 dedicated computers to the multiple computers to create a test specific  
4 environment.

5 Test results from the testing are gathered and/or sent to a party that creates  
6 the driver. Changes to the driver and/or hardware component may be made that  
7 optimize performance for the particular hardware configuration. The new driver  
8 and/or hardware component is then sent to the PC manufacturer or integrator. The  
9 new driver and/or hardware component is installed. New tests are performed with  
10 the new driver and/or hardware component. The new test results may be sent back  
11 to the party creating the drivers. A new driver and/or hardware component may or  
12 may not be sent to the PC manufacturer or integrator.

13 Creating an optimal driver or hardware component for each PC  
14 configuration may take numerous iterations of testing, gathering tests, sending  
15 tests, and retesting. When testing is involved and consumes considerable time,  
16 such as the case for graphics testing, there may not be the luxury to perform  
17 numerous iterations of testing.

18 Oftentimes, PC manufacturers and hardware components rely on a  
19 compromised “generic” driver that is used for all PC configurations. However,  
20 this may be unacceptable for particular PC configurations in that the associated  
21 hardware component may be inoperative or its operation considerably degraded  
22 for particular PC configurations. Substituting a customized driver or set of drivers  
23 for the generic driver may allow the same hardware component to properly  
24 function in the particular PC configuration. However, creating such drivers  
25 requires iterations of testing and retesting.

1 It is therefore desirable to provide a method and system of testing such as  
2 graphics testing from a central resource such as a server computer on various  
3 remote client computers (e.g., thin client PCs) without the need to install test tool  
4 applications on the remote client computers, and determine test results at either the  
5 client PC or the central resource.

6

7 **SUMMARY**

8 This disclosure describes computer graphics performance testing that  
9 includes establishing a session between a server computer having a test tool  
10 application and remote client computers. Unique instructions and data are sent to  
11 the remote client computer, a communication channel is created to allow testing  
12 information to be sent to the remote client computer, where the set of instructions  
13 and data is used to communicate through the communication channel.

14 In certain embodiments, individual tests are provided and testing  
15 information regarding each of the individual tests are sent over the communication  
16 channel and timed as to how long it takes for the testing information to be received  
17 by the remote client computer.

18

19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 Fig. 1 is a block diagram illustrating a system of remote client computers  
21 connected to a server computer by a network.

22 Fig. 2 is a block diagram illustrating the server computer, one remote client  
23 computer, and a communication channel to carry testing information between the  
24 server and remote client computers.

1 Fig. 3 is a flow chart illustrating a process for remotely testing the remote  
2 client computer using testing tools resident at the server computer.

3 Fig. 4 is a block diagram illustrating a general example of a computer that  
4 is used in accordance with the subject matter.

5

## 6 **DETAILED DESCRIPTION**

7 Fig. 1 illustrates a system 100 of computers connected by a network. In  
8 particular, a server computer 105 is connected to one or more remote client  
9 computers 110(1), 110(2) ... 110(N) through a network 115.

10 Server computer 105 and remote client computers 110(1), 110(2) ...  
11 110(N) are implemented with Microsoft® Windows® operating systems and may  
12 be configured with various hardware and software components. In particular,  
13 server computer 105 is configured with a test application program or programs  
14 that may be referred to as a test tool 107. Test tool 107 may be a graphics test  
15 application program that includes a list or series of individual tests. An example  
16 of test tool 107 is WinBench® by the Veritest® Company. The WinBench® test  
17 tool application comprises multiple individual tests that are run sequentially  
18 according to a list to test the graphics capabilities of a computer. Some of the tests  
19 performed include DirectDraw® tests which are directed to video processing from  
20 a PC central processing unit (CPU) to a video adaptor; disk inspection tests  
21 directed to hard disk utilization and CPU transfer rate; graphics display interface  
22 (GDI) testing directed to how graphics are displayed by the Microsoft®  
23 Windows® operating system; and WinMark® tests as defined by the Ziff-Davis  
24 Company directed to graphics accelerators, disk speed, and disk caching.

1 Testing information, such as individual graphics tests, from the test tool  
2 107 may be transferred to remote client computers 110(1), 110(2) ... 110(N).  
3 Remote client computers 110 may include thin client PCs that have minimal  
4 software application programs and rely on a server computer for application  
5 programs. Server computer 105 may operate as a server computer providing  
6 application programs, or another server computer (not shown) may provide such  
7 application programs.

8 Network 115 may be an “open” network that may include other computers  
9 and devices. It is contemplated that network 115 is not limited to providing a  
10 testing connection between server computer 105 and remote client computers  
11 110(1), 110(2) ... 110(N). In other words, network 115 may provide  
12 communications between the computers other than communications directed to  
13 testing. As such, remote client computers 110(1), 110(2) ... 110(N) may initiate  
14 or establish a session with server computer 105. Alternatively, server computer  
15 105 may establish a session with any one of remote client computers 110(1),  
16 110(2) ... 110(N). Establishing such a session may be performed by sending a  
17 request from server computer 105 to one of remote clients 110(1), 110(2) ...  
18 110(N) through network 105, or one of remote client computers 110(1), 110(2) ...  
19 110(N) sending a request to server computer 105.

20 Further, network 115 may be comprised of wide area networks (WAN) and  
21 various sub-networks, and include the Internet. Connections within network 115  
22 may include wired and wireless connections which allow server computer 105 and  
23 remote client computers 110(1), 110(2) ... 110(N) to couple to network 115. In  
24 certain instances particular computers may have a preferred connection to network  
25 105. For example, remote client computer 110(1) may be connected by a serial

1 IEEE 1394 “firewire” connection; remote client computer 110(2) may be  
2 connected through a universal serial bus (USB) connection; and remote client  
3 computer 110(N) may be connected through a wireless 802.11 connection.

4 Fig. 2 shows the server computer 105 and one of the remote client  
5 computers 110 in more detail. After a session is established between the remote  
6 client computer 110 and the server computer 105, the computers create a  
7 communication channel 205 between them. Communication channel 205 is  
8 established over network 105 of Fig. 1 to facilitate exchange of testing  
9 information, such as graphics testing, between the server computer 105 and the  
10 remote client computer 110.

11 Communication channel 205 may be implemented as a virtual channel  
12 which connects virtual channel connections (not shown) on server computer 105  
13 and remote client computer 110. Specifically, a virtual channel allows  
14 asynchronous transfer mode (ATM) cells to be transported between server  
15 computer 105 and remote client computer 110, where the ATM cells may  
16 represent testing information. In general a communication channel such as a  
17 virtual channel provides the ability to transfer information between two devices  
18 such as server computer 105 and remote client computer 110. In this example and  
19 as described further detail below, graphics testing information is sent from server  
20 computer 105 and received by remote client computer 110. In particular, tests  
21 (i.e., test information from tests) from test tool 107 are sent across communication  
22 channel 205, and the tests are timed as they received/Performed at either server  
23 computer 105 or remote client computer 110. For tests in a sequential list of tests,  
24 each test is timed as to how long the test is sent across communication channel  
25 205. A test in a sequentially ordered list of tests is not sent out unless the previous

1 test has been completely sent across communication channel 205. Therefore the  
2 time that a test takes to go across communication channel 205 is indicative of how  
3 well the test performed.

4 Server computer 105 includes an executable file software program called  
5 “serverRDP.exe” 210 which is a remote desktop protocol (RDP) program used to  
6 establish a session with remote client computer 110. The program  
7 “serverRDP.exe” may be resident in a startup folder in server computer 105 and  
8 initiated by remote client computer 110 when remote client computer 110 initiates  
9 a session. The software program “serverRDP.exe” 210 may be a Windows®  
10 based RDP used to communicate between and with thin client PCs. For example,  
11 server computer 105 is a thin client server computer accessible by remote client  
12 computer 110 which is a thin client PC.

13 Included in test tool application 107 is a server software module 215 that is  
14 comprised of a set of instructions and data. The set of instructions and data may  
15 be embodied as a dynamic link library (DLL) file. A DLL file is a set of programs  
16 which may be called up by a program that operates within a computer such as  
17 server computer 105. The DLL file allows other program operating within a  
18 computer to communicate with a device or another computer such as remote client  
19 computer 110. In this example, software module 215 provides test tool application  
20 107 the ability to transfer testing information to remote client computer 110 over  
21 the communication channel 205. Instructions and data of server software module  
22 215 may be in a register key in an operating system of server computer 105.

23 Server computer 105 further includes a timer 220 that is configured to time  
24 test and test information as they are sent across communications channel 205, and  
25

1 a test results log 225 configured to store test results for particular sessions with  
2 client computer 110.

3 Remote client computer 110 includes graphics hardware 230 that may  
4 include a graphics card, a display (i.e., a video monitor), video adapter cards, and a  
5 hard disk or storage device. In this example, graphics testing is performed on  
6 graphics hardware 230.

7 Remote client computer 110 includes an executable file software program  
8 called “clientRDP.exe” 235 which is a RDP program used to establish a session  
9 with server computer 105. Software program “clientRDP.exe” 235 may be a  
10 Windows® based RDP used to communicate to server computer 105, in particular  
11 “clientRDP.exe” is used to request instructions and data from server computer 105  
12 to be placed in a client software module 240 which is resident in remote client  
13 computer 110. Client software module 240 in particular receives unique  
14 instructions and data for each instance of a session that is established between  
15 server computer and remote client computer 110.

16 The software program “clientRDP.exe” 235 locates the instructions and  
17 data in software module 230 which are used to access test tool application 107.  
18 The set of instructions and data may include a DLL file. The set of instructions  
19 and data (i.e., DLL file) in client software module 240 is sent by server computer  
20 105 and received by remote client computer 110 through network 115 of Fig. 1.  
21 This may be in the form of a “DLL injection” which sends the required program  
22 (i.e., set of instructions and data) to the remote client computer 110 from the server  
23 computer 105. It is contemplated that the set of instructions and data can be sent  
24 and received prior to creating or establishing communication channel 205.

When initiated by remote client computer 110, “serverRDP.exe” 210 may set up a register key or keys that are used by an operating system of remote client computer 110 to load client software module 240 in an RDP client session module 245 resident in remote client computer 110. RDP client session module 245 may include a log file to store test times and other test results from a session between server computer 105 and remote client computer 110. RDP client session module 245 and its log file may be accessed for test times and test results.

Client software module 240 may time how long it takes for remote client computer 110 to connect to and log on to server computer 105 to establish a session. Timing may be performed by a resident timer 250. Once a session begins, “serverRDP.exe” 210 is initiated from the startup folder and initiates test tool application 107. Server software module 215 installs instructions and data from the register key in the operating system of server computer 105.

When test tool 107 is initiated, instructions in server software module 220 are sent to client software module 240 indicating the beginning of tests and that test information will be sent through communication channel 205.

Test tool 107 may include a list of individual tests that are ran sequentially one after the other. Each test has “begin” and “end” messages that are recognized by software modules 215 and 240. Remote client computer 110 may start timer 250 to time each individual test as test information is received. For example, in graphics testing, bit maps may be sent out as part of tests in test tool application 107. The bit maps are sent through communication channel 205 and timed from the start of receipt to the end of transmission.

In addition, server computer 105 may also perform timing through timer 220 of the tests by determining how long tests are sent to remote client computer

1       110. The tests of test tool 107 are run sequentially from the list and are timed by  
2 timer 220 and/or timer 250. Test times may be stored in a user accessible test  
3 results log 225 or a test log in RDP client session 245.

4              The beginning and ending of tests are provided to remote client computer  
5 110 through communication (i.e., messages sent and received) between server  
6 software module 215 and client software module 240. Specifically, server  
7 software module 215 is configured to look for the beginning and ending of tests  
8 (i.e., begin and end messages) of test application tool 107.

9              When test tool application 107 completes all tests, test tool application 107  
10 ends. The end of the testing is seen by server module 215 and provides the  
11 information to client software module 240 which informs remote client computer  
12 105 that testing has ended. Client software module 240 may time how long it  
13 takes to log off and end the session with server computer 105.

14 **Operation**

15              Fig. 3 is a flow chart illustrating a process 300 for remotely testing a remote  
16 client computer using a testing tool application resident at a server computer.  
17 Process 300 may be implemented with server computer 105 and remote client  
18 computer 110 as illustrated in Fig. 2. Process 300 illustrates blocks performed at  
19 the remote client computer side and the server computer side.

20              At block 305, the remote client computer establishes a testing session with  
21 the server computer. Establishing a testing may be initiated with a RDP such as  
22 “clientRDP.exe” 235 of Fig. 2 that sends a request to establish a testing session  
23 with the server computer. The remote client computer may time how long it takes  
24 to establish the testing session with the server computer.

25

1       Alternatively, at block 310, the server computer may establish a testing  
2 session with the remote client computer by sending a request. An RDP such as  
3 “serverRDP.exe” 210 may initiate the request for the server computer. The server  
4 computer may also time how long it takes to establish the testing session with the  
5 remote client computer.

6       At block 315, a set of instructions and data that may be in the form of a  
7 DLL is sent from the server computer. This set of instructions and data is stored at  
8 the server computer and may be unique to each particular remote client computer.  
9 The set of instructions and data may be placed in server software module 215  
10 which is configured to test tool 107 of Fig. 2. The set of instructions and data  
11 allow tests and test information from test tool 107 to be sent to the remote client  
12 computer.

13      At block 320, the set of instructions and data (e.g., DLL) is received by the  
14 remote client computer. The set of instructions and data may be stored in client  
15 software module 240 of Fig. 2 which allows the remote client compute to receive  
16 the tests and test information from test tool 107 of Fig. 2. As an example, the tests  
17 and test information in may be related to graphics testing.

18      At blocks 325 and 330, a communication channel is established or created  
19 with the remote client computer and the server computer. The communication  
20 channel may be a virtual channel as described above and illustrated as  
21 communication channel 205 of Fig. 2. Tests and test information are sent across  
22 the communication channel from the test tool and received at the remote client  
23 computer by a client software module. A unique channel identifier may be  
24 registered with the server computer that identifies the particular communication  
25 channel.

1 At block 335, the remote client computer may initiate the resident test tool  
2 of the server computer, through an RDP such as “clientRDP.exe” 235 of Fig. 2  
3 which may be placed in a start up folder as described above. The initialization  
4 command may be sent to an RDP in the server computer such as “serverRDP.exe”  
5 210 of Fig. 2.

6 At block 340, the server computer may start the resident test tool through  
7 an RDP such as “serverRDP.exe” 210 of Fig. 2. The test tool may include a list of  
8 individual tests that are ran sequentially one after the other in the list. The test tool  
9 may relate to graphics testing, and as an example may include a WinBench® test  
10 tool application as described above.

11 At block 345, the server computer begins sending a test, and indicates the  
12 start of each test, of a possible list of tests across through the established  
13 communication channel. A timer in server computer (e.g., timer 220 of Fig. 2)  
14 may be initiated as to when the test and/or test information is sent and time how  
15 long the test and/or test information takes to be sent. The times of the tests may be  
16 stored in a test log in the server computer.

17 At block 350, the remote client computer receives a test (i.e., the start of the  
18 test) and/or test information through the communication channel. A timer (e.g,  
19 timer 250 of Fig. 2) in the remote client computer is started when a start of the test  
20 is indicated, and stops when an end of the test is indicated by the server computer.  
21 Timing may be performed to determine how long it takes to receive the test and/or  
22 test information. The times of the test may be stored in log of a session file such  
23 as RDP client session file 245 of Fig. 2.

24 At block 355, a determination is made if the test is the last test in the list of  
25 tests of the test tool. The determination may be made at the remote client

1 computer and the server computer. The remote client computer, and in particular  
2 instructions received in block 320 allow the remote client computer to read begin  
3 and end messages for each test, and end of testing for the list of tests. The server  
4 computer looks to the resident test tool to indicate end of testing. If additional  
5 tests are to be performed (following the “NO” branch of block 355), block 345 is  
6 performed.

7 If there no additional tests to be performed (following the “YES” branch of  
8 block 355), blocks 360 and 365 may be performed concurrently or exclusively. At  
9 Block 360, a logoff takes place that ends the testing session for the remote client  
10 computer. Timing may be performed at the client computer as to how long it takes  
11 to log off the session. At block 365, a logoff takes place that ends testing for the  
12 server computer. Timing may also be performed at the server computer to  
13 determine how long it takes to log off the session.

14 **Exemplary Computer Environment**

15 The subject matter is described in the general context of computer-  
16 executable instructions, such as program modules, being executed by a computer  
17 such as server computer 105 and remote client computers 110(1), 110(2) ...  
18 110(N) of Fig. 1. Generally, program modules include routines, programs, objects,  
19 components, data structures, etc. that perform particular tasks or implement  
20 particular abstract data types. Moreover, those skilled in the art will appreciate  
21 that the subject matter may be practiced with other computer system  
22 configurations, including hand-held devices, multiprocessor systems,  
23 microprocessor-based or programmable consumer electronics, network PCs,  
24 minicomputers, mainframe computers, and the like. In a distributed computer  
25

1 environment, program modules may be located in both local and remote memory  
2 storage devices.

3 Fig. 4 shows a general example of a computer 430 that is used in  
4 accordance with the subject matter. Computer 430 is shown as an example of a  
5 computer that can perform the functions of a server computer or a remote client  
6 computer. Computer 430 includes one or more processors or processing units 432,  
7 a system memory 434, and a bus 436 that couples various system components  
8 including the system memory 434 to processors 432. Processors 432 may initiate  
9 or configured to initiate a session with another computer. For example, as server  
10 computer 105, the program “serverRDP.exe” 210 of Fig. 2 instructs processor 432  
11 to initiate or allow initiating of a session. As remote client computer 110, the  
12 program “clientRDP.exe” 235 instructs processor 432 to initiate or allowing  
13 initiating of a session.

14 The bus 436 represents one or more of any of several types of bus  
15 structures, including a memory bus or memory controller, a peripheral bus, an  
16 accelerated graphics port, and a processor or local bus using any of a variety of  
17 bus architectures. The system memory includes read only memory (ROM) 438  
18 and random access memory (RAM) 440.

19 Programs such as “clientRDP.exe” 225 or “serverRDP.exe” 210, test tool  
20 application 215, RDP client session module 235, and test results log 240 of Fig. 2  
21 may be stored in ROM 438 or RAM 440. ROM 438 or RAM 440 may be  
22 configured with registers or storage locations to indicate various sets of  
23 instructions and data (e.g., DLL files) used for various sessions between server  
24 computer 105 and remote client computers 110(1), 110(2) ... 110(N).

25

1 A basic input/output system (BIOS) 442, containing the basic routines that  
2 help to transfer information between elements within computer 430, such as  
3 during start-up, is stored in ROM 438. Computer 430 further includes a hard disk  
4 drive 444 for reading from and writing to a hard disk, not shown, a magnetic disk  
5 drive 446 for reading from and writing to a removable magnetic disk 448, and an  
6 optical disk drive 450 for reading from or writing to a removable optical disk 452  
7 such as a CD ROM or other optical media. The hard disk drive 444, magnetic disk  
8 drive 446, and optical disk drive 450 are connected to the bus 436 by an SCSI  
9 interface 454 or some other appropriate interface. The drives and their associated  
10 computer-readable media provide nonvolatile storage of computer readable  
11 instructions, data structures, program modules and other data for computer 430.

12 Although the exemplary environment described herein employs a hard disk,  
13 a removable magnetic disk 448 and a removable optical disk 452, it should be  
14 appreciated by those skilled in the art that other types of computer readable media  
15 which can store data that is accessible by a computer, such as magnetic cassettes,  
16 flash memory cards, digital video disks, random access memories (RAMs) read  
17 only memories (ROM), and the like, may also be used in the exemplary operating  
18 environment.

19 A number of program modules may be stored on the hard disk, magnetic  
20 disk 448, optical disk 452, ROM 438, or RAM 440, including an operating system  
21 458, one or more application programs 460, other program modules 462, and  
22 program data 464.

23 A user may enter commands and information into computer 430 through  
24 input devices such as keyboard 466 and pointing device 468. Other input devices  
25 (not shown) may include a microphone, joystick, game pad, satellite dish, scanner,

1 or the like. These and other input devices are connected to the processing unit 432  
2 through interface 470 that is coupled to bus 436. Monitor 472 or other type of  
3 display device is also connected to bus 436 via an interface, such as video adapter  
4 474.

5 Computer 430 operates in a networked environment using logical  
6 connections to one or more remote computers, such as a remote computer 476.  
7 The remote computer 476 may be another personal computer, a server, a router, a  
8 network PC, a peer device or other common network node, and typically includes  
9 many or all of the elements described above relative to computer 430, although  
10 only a memory storage device 478 has been illustrated in Fig. 4. The logical  
11 connections depicted in Fig. 4 include a local area network (LAN) 480 and a wide  
12 area network (WAN) 482. Such networking environments are commonplace in  
13 offices, enterprise-wide computer networks, intranets, and the Internet.

14 When used in a LAN networking environment, computer 430 is connected  
15 to the local network 480 through a network interface or adapter 484. When used  
16 in a WAN networking environment, computer 430 typically includes a modem 486  
17 or other means for establishing communications over the wide area network 482,  
18 such as the Internet. The modem 486, which may be internal or external, is  
19 connected to the bus 436 via a serial port interface 456. In a networked  
20 environment, program modules depicted relative to the personal computer 430, or  
21 portions thereof, may be stored in the remote memory storage device. It will be  
22 appreciated that the network connections shown are exemplary and other means of  
23 establishing a communications link between the computers may be used.

24 Generally, the data processors of computer 430 are programmed by means  
25 of instructions stored at different times in the various computer-readable storage

1 media of the computer. Programs and operating systems are typically distributed,  
2 for example, on floppy disks or CD-ROMs. From there, they are installed or  
3 loaded into the secondary memory of a computer. At execution, they are loaded at  
4 least partially into the computer's primary electronic memory.

5 The subject matter described herein includes these and other various types  
6 of computer-readable storage media when such media contain instructions or  
7 programs for implementing the steps described below in reference to Fig. 4 in  
8 conjunction with a microprocessor or other data processor.

9 The subject matter also includes the computer itself when programmed  
10 according to the methods and techniques described below. Furthermore, certain  
11 sub-components of the computer may be programmed to perform the functions  
12 and steps described below. The subject matter includes such sub-components  
13 when they are programmed as described. In addition, the subject matter described  
14 herein includes data structures, described below, as embodied on various types of  
15 memory media.

16 For purposes of illustration, data, programs and other executable program  
17 components, such as the operating system are illustrated herein as discrete blocks,  
18 although it is recognized that such programs and components reside at various  
19 times in different storage components of the computer, and are executed by the  
20 data processor(s) of the computer.

21 Although the invention has been described in language specific to structural  
22 features and/or methodological acts, it is to be understood that the invention  
23 defined in the appended claims is not necessarily limited to the specific features or  
24 acts described. Rather, the specific features and acts are disclosed as exemplary  
25 forms of implementing the claimed invention.